51CTO.com
技术成就梦想

我 们 只 谈 开 发

开发月刊

Development Monthly

2011年8月
总第005期

jQuery Mobile

高手必备十大技巧和代码片段

JRuby和Java 7我们可以期待什么

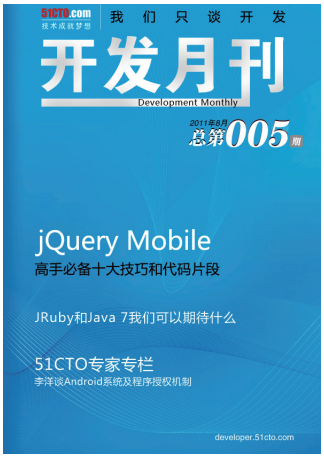
51CTO专家专栏

李洋谈Android系统及程序授权机制

developer.51cto.com

目录

2011.08 Contents



编程排行 Billboard

3 8月编程榜:总统为名的语言新贵Lua

专题报道 《jQuery移动开发指南》

7 jQuery Mobile高手必备的技巧和代码片段

9 jQuery设计思想

12 jQuery是如何工作的

14 10个出色的jQuery Mobile插件

技术热点 Tech Focus

5 Web开发工具选择和使用情况调查

16 我们要做会写代码的项目经理

17 五分钟理解Python元类

19 HTML 5遭欧盟网络安全机构警告

20 程序员如何与非程序员一起工作

21 Scala登陆.NET平台

23 Java 8整装待发 图谋云计算

25 程序员育儿新工具Small Basic

26 教你五招，提升MySQL可扩展性

27 几点建议，让Redis发挥更大作用

29 YiBo微博:程序员创业之梦

移动开发 Mobile Dev

31 李洋谈Android安全框架

特别推荐 Best topic

32 数据库的“轮回”

51CTO.com 技术成就梦想

出版方:北京无忧创想信息技术有限公司 责任编辑:彭凡

编辑:周少男 陈贻新 边艾婧 封面设计:魏峰

《开发月刊》欢迎大家踊跃投稿,来稿请注明“开发月刊投稿”并发送至 pengfan@51cto.com

出版日期:2011 年 8 月 14 日(每月 14 日出版)

8月编程榜:总统为名的语言新贵Lua

8 月 4 日，Tiobe 公布了最新的 8 月编程语言排行榜，和 7 月榜单相比，变化不到。1 至 10 名中，PHP 和 C# 交换位置，其他不变。本期 51CTO 编程语言排行榜便以 Lua 语言为主题，介绍这款以巴西总统的名字命名的开发语言新贵。

8 月初，Tiobe 公布了最新的 8 月编程语言排行榜，和 7 月榜单相比，最大变动就数微软的 F# 了，F# 首次进入 TOP20。1 至 10 名中，PHP 重返第 4，C# 则到了第 5，其他不变。不过看看最近人气上升的 F# ，就没有什么好奇怪的了。作为一种不错的程序设计语言，微软最新的 Visual Studio 2010 中也包含了 F#。

Position Aug 2011	Position Aug 2010	Delta in Position	Programming Language
1	1	=	Java
2	2	=	C
3	3	=	C++
4	4	=	PHP
5	6	↑	C#
6	9	↑↑↑	Objective-C
7	5	↓↓	(Visual) Basic
8	7	↓	Python
9	8	↓	Perl
10	11	↑	JavaScript
11	23	↑↑↑↑↑↑↑↑↑↑	Lua
12	12	=	Ruby
13	10	↓↓↓	Delphi/Object Pascal
14	16	↑↑	Lisp
15	22	↑↑↑↑↑↑↑	Transact-SQL
16	28	↑↑↑↑↑↑↑↑↑↑	Ada
17	19	↑↑	RPG (OS/400)
18	17	↓	Pascal
19	46	↑↑↑↑↑↑↑↑↑↑	F#
20	-	=	Assembly*

上图是前 20 名编程语言排行

2011 年 6 月、7 月的 Tiobe 编程语言排行榜都介绍了移动开发语言 Lua，其中 7 月编程语言排行榜以年度语言预测为主题，直言 Lua 是 2011 年度唯一有机会挑战 Objective-C 的语言。但是 Lua 到底是什么语言呢？本期 51CTO 编程语言排行榜便以 Lua 语言为主题，介绍这款以巴西总统的名字命名的移动开发语言新贵。



Lua 是一个被设计成用于扩展 C/C++ 编程的轻量级脚本语言，它的目标是成为一个很容易嵌入其它语言中使用的语言。Lua 让 C/C++ 编写的代码调用 Lua 脚本，同时 C/C++ 函数也能够在 Lua 脚本中调用。与 C/C++ 代码的交互性，让 Lua 在应用程序开发中有着广泛的应用基础。很多应用程序使用 Lua 作为自己的嵌入式脚本语言，以此来实现可配置性、可扩展性。

Lua 名字的由来

小巧的 Lua 语言来自于巴西里约热内卢天天主教大学的实验性项目。当初巴西里约热内卢天主教大学的师生之所以给这款语言起名为 Lua，

8 月编程语言排行榜：总统为名的语言新贵 Lua II

是为了向巴西著名政治家、前任总统路易斯 · 伊纳西奥 · 卢阿 · 达席尔瓦致敬。卢阿总统是是巴西历史上学历最低，但是支持率最高的总统，深受人民爱戴，作为巴西国家电子领域最重要的人才培养基地的里约热内卢天主教大学计算机研究生院把他们在计算机世界中最重要成就“Lua”语言与他们总统同名，可见他们对 Lua 语言给予了多么大的希望。



Lua 语言在移动开发中的成功

2011 年以前中国人对 Lua 语言的老家里约热内卢可以说是基本不了解，不过一部以小鸟爱情为主题的动画片《里约大冒险》让我们了解了里约热内卢的美丽风光。同样一个小鸟保卫鸟蛋的游戏《愤怒的小鸟》让 Lua 语言风靡开发者的世界。

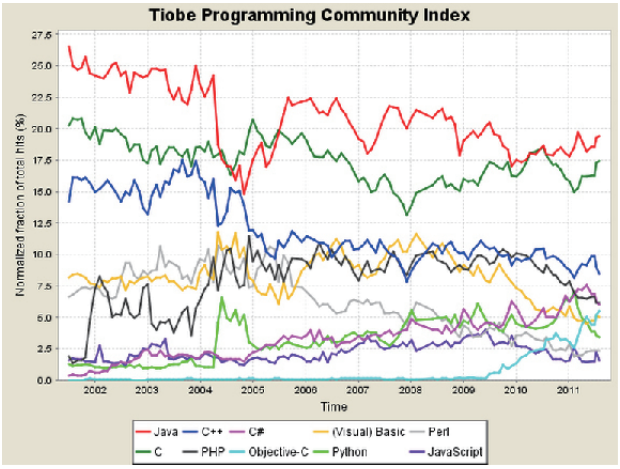


Lua 的技术特点

目前 Lua 语言的最新版本是 5.2 Beta，Lua 也能对面向对象编程，函数式编程，数据驱动式编程提供很好的支持。作为一个扩展式语言，Lua 没有 "main" 程序的概念：它只能嵌入一个宿主程序中工作，这个宿主程序被称作 embedding program 或简称为 host。宿主程序可以通过调用函数执行一小段 Lua 代码，可以读写 Lua 变量，可以注入 C 函数让 Lua 代码调用。这些扩展的 C 函数，可以大大的扩展了 Lua 可以处理事务的领域，这样就可以订制出各种语言，而它们共享一个统一的句法格式的框架。

很多网站宣传 Lua 语言作为脚本开发了《魔兽世界》，其实 Lua 语言只是开发了《魔兽世界》的插件，几乎所有的《魔兽世界》插件都是使用解释性的脚本 Lua 编写，核心的 C++ 代码只占全部代码中的很小一部分比重。在传统平台上 Lua 最流行的应用场景是在游戏中作为处理逻辑的脚本，即用来编写游戏的任务脚本，以实现游戏的可配置性和可扩展性。在游戏中 Lua 脚本的加载不需重启服务器，尽管脚本的运行效率低于编译好的二进制代码，但目前 CPU 速度已完全能够让 Lua 脚本迅速解释执行。这也正是包括《魔兽世界》、《战锤 40K》、《仙境传说》、《博德之门》和《轩辕剑外传汉之云》在内的众多游戏选择 Lua 开发插件的原因。（本文作者：周少男）

前 10 名编程语言走势图



查看原文：

http://developer.51cto.com/art/201108/281135.htm

Web开发工具选择和使用情况调查

51CTO 调研中心

目前，基于各种开发平台或语言的 Web 开发工具层出不穷，其热度、知名度和排名几经更迭。Web 开发工具种类的丰富增加了选择的范围，同时也增加了选择成本。从学习的时间成本到就业前景考虑，哪些开发工具目前使用更为广泛？应该重点学习哪种开发工具？一直是困扰着准备或已经进入到 Web 开发行业并立志于此的人们。

完整调查报告请点击：

http://down.51cto.com/data/232581

在线阅读地址：

http://book.51cto.com/art/201108/282240.htm

各大 IT 技术网站、技术论坛等对各种开发工具的界面美观、结构化、直观易学、通用性等的看法和观点也是见仁见智，有时甚至大相径庭。可见，评价“哪种开发工具最好”是一个费力不讨好的工作，因为根本没有完美无缺的开发工具。但是，来自不同团队的程序员、技术经理、技术总监，都对 51CTO 表达过一个观点，那就是使用合适的开发工具，会对团队的开发效率产生明显作用。

51CTO 组织了本次调研活动，希望对开发工具的使用情况进行更加深入的了解。很显然，通过本次调研我们不能给出一个“哪种开发工具更好”的肯定答案；但是，我们可以从这次活动中了解到不同职位、不同工作年限的开发人员，他们选择和使用开发工具的情况；以及不同规模的开发团队，他们对开发工具的选择又是怎样，从而协助我们为自己的开发团队选择正确的工具。

Web 开发者指的是从事网站整体开发的程序员，他们由前端开发工程师、研发工程师、技术主管、项目经理和架构师等职位组成。51CTO 经过一个月的细致调查，在海量调查样本的基础上完成了《2011 年中国 Web 开发工具选择和使用调查报告》，让你走进身边的 Web 开发者，了解他们的工作习惯。

这是年轻人的事业

互联网是全新的行业，而从事 Web 开发的程序员更是年轻人的天下。根据 51CTO 调查，一半多的 Web 开发者年龄在 23 到 26 岁，这个年龄正好是 22 岁大学毕业后的事业期。处于 27 岁到 30 岁事业上升期的 Web 开发者有 21%，而 30 岁以上而立之年的 Web 开发者仅占 5.8%。

这样的年龄段充分说明 Web 开发者大多数还处于青壮年，能适应超长时间加班和快速的项目反应。相信各位网站技术人员，经常会有半夜被叫醒处理 404 错误等问题的经历。

23 到 26 岁一般在北上广深等城市还不具备结婚的条件，这也比较方便互联网公司将 Web 开发者拉到郊外，进行封闭集中开发。

至于 18 到 22 岁的 Web 开发者大多是个人网站站长，他们基本是从个人兴趣出发从事 Web 开发工作。

当 Web 开发者到达 30 岁这个事业里程碑时，基本上都要晋升到技术主管或者项目经理层次才能继续从事 Web 开发工作。其余的 Web 开发者将经历痛苦的职业转型，这也是其他程序员不得不经历的过程。

Bear Bibeault
Yehuda Katz

Covers jQuery 1.4 and jQuery UI 1.8



jQuery IN ACTION

SECOND EDITION

jQuery 是继 prototype 之后又一个优秀的 Javascript 框架。它是轻量级的 js 库，兼容 CSS3，还兼容各种浏览器。

为了让移动设备也能用上 jQuery，jQuery 开发团队发布了 jQuery Mobile。jQuery Mobile 支持全球主流的移动平台和浏览器，让移动 Web 开发更加方便。

更多内容请查看“jQuery Mobile”专题：

http://mobile.51cto.com/others-277308.htm

jQuery 从入门到精通

http://developer.51cto.com/art/201005/202450.htm

51CTO 开发频道寄语

jQuery Mobile高手必备的技巧和代码片段

本文不是旨在全面介绍使用jQuery Mobile库的入门知识，会忽略掉一些相当浅显的东西，而是直接探讨一些相当棘手的东西：51CTO会向大家介绍jQuery Mobile高手必备的技巧和代码片段。



与任何新技术一样，常常难就难在如何开始入手。有鉴于此，我们整理出了与jQuery Mobile库有关的我认为最便利的一些技巧、方法和代码片段。

由于本文不是旨在全面介绍使用jQuery Mobile库的入门知识，会忽略掉一些相当浅显的东西，而是直接探讨一些相当棘手的东西。

完整的基本页面

我一再发现自己需要为基本页面建立完整的标记。正因为如此，下面是创建一个基本页面所需要的全部代码。

```
<!DOCTYPE html><html><head>
<title>Page Title</title>
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.5.min.js"></script>
<script type="text/javascript" src="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.js"></script> </head>
<body><div data-role="page" id="home">
<div data-role="header"><h1>Header</h1></div>
<div data-role="content">
<p>Content goes here</p></div>
<div data-role="footer"><h4>Footer</h4></div>
</div></body></html>
```

何处添加额外的 jQuery 调用

当我开始使用jQuery的这个优秀扩展时，立即发现自己需要在触发移动插件之前改动页面上的内容。结果发现，建议的解决办法就是只要把传统的jQuery调用放在装入移动插件的引用之前。这样，你的jQuery命令就有机会在库装入之前运行。下面是要遵照的模式：

```
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.5.min.js"></script>
<script>
$(document).ready(function()
{// Your jQuery commands go here before the mobile reference }); </script>
<script type="text/javascript" src="http://code.jquery.com/mobile/1.0a3/jquery.mobile-1.0a3.min.js"></script>
```

阻止一些关键项目被截断

库的一项特性（或者是缺点，就看你的需要了）在于，它能够智能化地截断很长的项目，以适应用户界面元素。

我发现在两种情况下这可能很烦人。首先在列表项目中，我更喜欢看到全部文本。其次是在脚注文本中。一旦你在这里有好多个字符，它似乎就会开始用“…”来截断。使用这个简单的CSS来覆盖这两种默认情况。

对于列表项目：

jQuery Mobile 高手必备的技巧和代码片段 II

```
body .ui-li .ui-li-desc { white-space: normal; }
```

对于脚注内容：

```
body .ui-footer .ui-title { white-space: normal;}
```

同时禁用所有链接的 AJAX 导航

尽管 AJAX 导航技术很出色，但有时候你倒希望禁用它。使用这一段 jQuery 代码，告诉 jQuery Mobile 库不要使用 AJAX 导航。

对于列表项目：

```
body .ui-li .ui-li-desc { white-space: normal; }
```

对于脚注内容：

```
body .ui-footer .ui-title { white-space: normal;}
```

同时禁用所有链接的 AJAX 导航

尽管 AJAX 导航技术很出色，但有时候你倒希望禁用它。使用这一段 jQuery 代码，告诉 jQuery Mobile 库不要使用 AJAX 导航。

把它放在页面标题中 jQuery Mobile 库引用的后面。换句话说，库必须已经在该代码被引用之前装入完毕。

```
<script>
$(document).ready(function()
{// disable ajax nav
$.mobile.ajaxLinksEnabled = false; });
</script>
```

创建弹出式对话框

jQuery Mobile 库的一项便利特性就是内置的弹出内容或对话框特性。创建这项便利的特性简单得很。基本上只要添加需要链接的属性，如下所示：data-rel="dialog"。

注意两点。首先，目标页面必须是完整的 jQuery Mobile 页面，如第一个技巧所概述。

其次，这只适用于外部页面；它必须是完整的单独页面才能正常工作。

```
<a href="#pop.html" data-rel="dialog">Pop up!</a>
```

阻止一些关键项目被截断

库的一项特性（或者是缺点，就看你的需要了）在于，它能够智能化地截断很长的项目，以适应用户界面元素。

我发现在两种情况下这可能很烦人。首先在列表项目中，我更喜欢看到全部文本。其次是在脚注文本中。一旦你在这里有好多个字符，它似乎就会开始用“…”来截断。使用这个简单的 CSS 来覆盖这两种默认情况。

对于列表项目：

```
body .ui-li .ui-li-desc { white-space: normal; }
```

对于脚注内容：

```
body .ui-footer .ui-title { white-space: normal; }
```

结论

jQuery Mobile 库处理起来很方便。只要稍微下一点工夫，它就能获得出色的效果。考虑到它目前仍处于测试阶段，所以它起步不错。

但愿你在不断钻研这个新的库时，这些快速技巧能让你不断前进。

更多内容，请阅读原文，链接：

http://mobile.51cto.com/hot-276160.htm

（文章编译：布加迪）■

为了让移动设备也能用上 jQuery，jQuery 开发团队发布了 jQuery Mobile。

jQuery Mobile 支持全球主流的移动平台和浏览器，让移动 Web 开发更加方便。

更多内容请查看“jQuery Mobile”专题：

http://mobile.51cto.com/others-277308.htm

jQuery设计思想

作者 阮一峰

jQuery 是目前使用最广泛的 JavaScript 函数库。据统计，全世界排名前 100 万的网站，有 46% 使用 jQuery，远远超过其他库。微软公司甚至把 jQuery 作为他们的官方库。对于网页开发者来说，学会 jQuery 是必要的。因为它让你了解业界最通用的技术，为将来学习更高级的库打下基础，并且确实可以很轻松地做出许多复杂的效果。虽然 jQuery 上手简单，比其他库容易学会，但是要全面掌握，却不轻松。因为它涉及到网页开发的方方面面，提供的各种方法和内部变化有上千种之多。初学者常常感到，入门很方便，提高很困难。

下面就是我的笔记，它应该是目前网上不多的 jQuery 中文教程之一。你只需要一点 JavaScript 语言的基本知识，就能看懂它，在最短的时间里，掌握 jQuery 的所有主要方面（除了 ajax 和插件开发）。

jQuery 设计思想

一、选择网页元素 jQuery 的基本设计和主要用法，就是 " 选择某个网页元素，然后对其进行某种操作 "。这是它区别于其他函数库的根本特点。

使用 jQuery 的第一步，往往就是将一个选择表达式，放进构造函数 jQuery()（简写为 $），然后得到被选中的元素。

选择表达式可以是 CSS 选择器：

```
$(document)// 选择整个文档对象
$('#myId')// 选择 ID 为 myId 的网页元素
$('div.myClass')// 选择 class 为 myClass 的 div 元素
$('input[name=first]')// 选择 name 属性等于 first 的 input 元素
```

也可以是 jQuery 特有的表达式：

```
$('a:first')// 选择网页中第一个 a 元素
$('tr:odd')// 选择表格的奇数行
$('#myForm :input')// 选择表单中的 input 元素
$('div:visible') // 选择可见的 div 元素
$('div:gt(2)')// 选择所有的 div 元素，除了前三个
$('div:animated')// 选择当前处于动画状态的 div 元素
```

二、改变结果集

如果选中多个元素，jQuery 提供过滤器，可以缩小结果集：

```
$('div').has('p'); // 选择包含 p 元素的 div 元素
$('div').not('.myClass'); // 选择 class 不等于 myClass 的 div 元素
$('div').filter('.myClass'); // 选择 class 等于 myClass 的 div 元素
$('div').first(); // 选择第 1 个 div 元素
$('div').eq(5); // 选择第 6 个 div 元素
```

有时候，我们需要从结果集出发，移动到附近的相关元素，jQuery 也提供了在 DOM 树上的移动方法：

```
$('div').next('p'); // 选择 div 元素后面的第一个 p 元素
$('div').parent(); // 选择 div 元素的父元素
$('div').closest('form'); // 选择离 div 最近的那个 form 父元素
$('div').children(); // 选择 div 的所有子元素
$('div').siblings(); // 选择 div 的同级元素
```

三、链式操作

选中网页元素以后，就可以对它进行某种操作。jQuery 允许将所有操作连接在一起，以链条的形式写出来，比如：

```
$('div').find('h3').eq(2).html('Hello');
```

分解开来，就是下面这样：

```
$('div')// 找到 div 元素
.find('h3')// 选择其中的 h3 元素
.eq(2)// 选择第 3 个 h3 元素
.html('Hello'); // 将它的内容改为 Hello
```

这是 jQuery 最令人称道、最方便的特点。它的原理在于每一步的 jQuery 操作，返回的都是一个 jQuery 对象，所以不同操作可以连在一起。

jQuery 还提供了 .end() 方法，使得结果集可以后退一步：

```
$('div')
.find('h3')
.eq(2)
.html('Hello')
.end()// 退回到选中所有的 h3 元素的那一步
.eq(0)// 选中第一个 h3 元素
.html('World'); // 将它的内容改为 World
```

四、元素的操作：取值和赋值

操作网页元素，最常见的需求是取得它们的值，或者对它们进行赋值。

jQuery 使用同一个函数，来完成取值 (getter) 和赋值 (setter)。到底是取值还是赋值，由函数的参数决定。

```
$('h1').html(); //html() 没有参数，表示取出 h1 的值 $('h1').html('Hello'); //html() 有参数 Hello，表示对 h1 进行赋值
```

常见的取值和赋值函数如下：

.html()
.text() 取出或设置 text 内容
.attr() 取出或设置某个属性的值
.width() 取出或设置某个元素的宽度
.height() 取出或设置某个元素的高度
.val()
取出或设置 html 内容 取出某个表单元素值

需要注意的是，如果结果集包含多个元素，那么赋值的时候，将对其中所有的元素赋值；取值的时候，则是只取出第一个元素的值 (.text() 例外，它取出所有元素的 text 内容)。

五、元素的操作：移动

如果要移动选中的元素，有两种方法：一种是直接移动该元素，另一种是移动其他元素，使得目标元素达到我们想要的位置。

假定我们选中了一个 div 元素，需要把它移动到 p 元素后面。

第一种方法是使用 .insertAfter()，把 div 元素移动 p 元素后面：

```
$('div').insertAfter('p');
```

第二种方法是使用 .after()，把 p 元素加到 div 元素前面：

```
$('p').after('div');
```

表面上看，这两种方法的效果是一样的，唯一的不同似乎只是操作视角的不同。但是实际上，它们有一个重大差别，那就是返回的元素不一样。第一种方法返回 div 元素，第二种方法返回 p 元素。你可以根据需要，选择到底使用哪一种方法。

使用这种模式的操作方法，一共有四对

.insertAfter() 和 .after()：在现存元素的外部，从后面插入元素
.insertBefore() 和 .before()：在现存元素的外部，从前面插入元素
.appendTo() 和 .append()：在现存元素的内部，从后面插入元素
.prependTo() 和 .prepend()：在现存元素的内部，从前面插入元素

六、元素的操作：复制、删除和创建

复制元素使用 .clone()。删除元素使用 .remove() 和 .detach()。两者的区别在于，前者不保留被删除元素的事件，后者保留，有利于重新插入文档时使用。

jQuery 设计思想 III

前者不保留被删除元素的事件，后者保留，有利于重新插入文档时使用。

清空元素内容（但是不删除该元素）使用 .empty()。

创建新元素的方法非常简单，只要把新元素直接传入 jQuery 的构造函数就行了：

```
$('<p>Hello</p>');

$('<li class="new">new list item</li>');
$('ul').append('<li>list item</li>');
```

七、工具方法

除了对选中的元素进行操作以外，jQuery 还提供一些工具方法（utility），不必选中元素，就可以直接使用。

如果你懂得 Javascript 语言的继承原理，那么就能理解工具方法的实质。它是定义在 jQuery 构造函数上的方法，即 jQuery.method()，所以可以直接使用。而那些操作元素的方法，是定义在构造函数的 prototype 对象上的方法，即 jQuery.prototype.method()，所以必须生成实例（即选中元素）后使用。如果不理解这种区别，问题也不大，只要把工具方法理解成，是像 javascript 原生函数那样，可以直接使用的方法就行了。

常用的工具方法有以下几种：

$.trim() 去除字符串两端的空格。
$.each() 遍历一个数组或对象。
$.inArray() 返回一个值在数组中的索引位置。如果该值不在数组中，则返回 -1。
$.grep() 返回数组中符合某种标准的元素。
$.extend() 将多个对象，并到第一个对象。
$.makeArray() 将对象转化为数组。
$.type() 判断对象的类别（函数对象、日期对象、数组对象、正则对象等等）。
$.isArray() 判断某个参数是否为数组。
$.isEmptyObject() 判断某个对象是否为空（不含有任何属性）。
$.isFunction() 判断某个参数是否为函数。
$.isPlainObject() 判断某个参数是否为用 "{}" 或 "new Object" 建立的对象。
$.support() 判断浏览器是否支持某个特性。

八、事件操作（详细请点击：http://developer.51cto.com/art/201107/278647_1.htm）

九、特殊效果

jQuery 允许对象呈现某些特殊效果。

```
$('h1').show(); // 展现一个 h1 标题
```

常用的特殊效果如下：

.fadeIn() 淡入
.fadeOut() 淡出
.fadeTo() 调整透明度
.hide() 隐藏元素
.show() 显示元素
.slideDown() 向下展开
.slideUp() 向上卷起
.slideToggle() 依次展开或卷起某个元素
.toggle() 依次展示或隐藏某个元素

除了 .show() 和 .hide()，所有其他特效的默认执行时间都是 400ms，但是你可以改变这个设置。

```
$('h1').fadeIn(300); // 300 毫秒内淡入
$('h1').fadeOut('slow'); // 缓慢地淡出
```

在特效结束后，可以指定执行某个函数。

```
$('p').fadeOut(300, function() { $(this).remove(); });
```

更复杂的特效，可以用 .animate() 自定义。

```
$('div').animate(
  {left : "+=50", // 不断右移
    opacity : 0.25 // 指定透明度 },
    300, // 持续时间
function() { alert('done!'); } // 回调函数 );
```

.stop() 和 .delay() 用来停止或延缓特效执行。

$.fx.off 如果设置为 true，关闭所有网页特效。

http://developer.51cto.com/art/201107/278647.htm ■

jQuery是如何工作的

最近发现jQuery很成熟了，我看了jQuery官方的document，这里先提供入门章节内容。主要介绍jQuery是如何工作的。



这段主要教授你如何开始使用jQuery. 如果你还没有一个测试页面，请先从按照下面的代码创建一个HTML页面。

```
<!doctype html>
<html>
 <head>
   <meta charset="utf-8">
   <title>Demo</title>
 </head>
 <body>
   <a href="http://jquery.com/">jQuery</a>
   <script src="jquery.js"></script>
   <script>
   </script>
 </body>
</html>
```

编辑Script标签的src属性，指向你的jQuery.js属性。例如，如果你的jQuery.js和HTML文件在同一文件夹下：

```
<script src="jquery.js"></script>
```

页面加载完成时运行代码

很多Javascript程序员都要做的第一件事情，就是加这样的一些代码到他们的程序：

```
window.onload = function(){ alert("welcome");}
```

这段代码将会在页面加载成功时被正确执行，但是，这段代码的问题是Javascript代码将会在所有的图片信息被全部下载完成后执行（这里包含了广告栏）。

这里使用首先Window.onload的原因是HTML在你最初运行这段代码的时候，HTML文档并没有完全加载。

jQuery有一个很简单的声明来检查文档，和判断文档是否已经准备完成，这里称之为ready event：

```
$(document).ready(function(){
  // Your code here   });
```

我们在ready event中添加一个超链接点击的处理方法。

```
$(document).ready(function(){
  $("a").click(function(event){
    alert("Thanks for visiting!");   });  });
```

保存HTML文件，在浏览器中刷新此文件，点击网页中的超链接，在连接到指定页面之前首先会弹出一个提示框。

在点击和其他的很多事件中，你可以在这里使用event.preventDefault()来阻止其进行默认行为。

```
$(document).ready(function(){
  $("a").click(function(event){
    alert("As you can see, the link no longer
took you to jquery.com");
    event.preventDefault();   });  });
```

完整示例

这里提供一个完整的HTML文件的示例来演示之前我们讲述的内容。注意这里连接到了Google's CDN来加载jQuery的核心库。

jQuery 是如何工作的 II

你最好将它放在一个单独文件中，并且在 Head 标签中完成对它的加载。

```
<!DOCTYPE html>
<html lang="en"> <head>
<meta charset="utf-8">
<title>jQuery demo</title>
</head>
<body>
<a href="http://jquery.com/">jQuery</a>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("a").click(function(event){
alert("As you can see, the link no longer took you to jquery.com");
event.preventDefault(); }); });
</script>
</body> </html>
```

增加和删除 css 样式

重要：接下来的 jQuery 示例需要放在 ready event 内，这样在文档准备结束后立即执行。

我们另一种常常做的事情，便是增加（或者删除）一个 css 样式。

首先我们在 <head> 标签内增加一些样式：

```
<style>
a.test { font-weight: bold; }
</style>
```

然后在代码内使用 addClass 方法

```
$("a").addClass("test");
```

所有超链接会被加粗。

要删除此样式，只需要使用 remove class 方法。（允许添加多个样式到 html)

```
$("a").removeClass("test");
```

特效

jQuery 提供了一组特效，使用这些特效可以让你的网站脱颖而出，你只需要将上面的测试文件添加如下代码就可以进行测试：

```
$("a").click(function(event){
event.preventDefault();
$(this).hide("slow");
});
```

如果你点击超链接，你所点击的超级链接会被隐藏。

回调函数

回调函数是指父方法运行通过后传递参数并且进行调用的函数。回调函数的特殊之处在父函数可以在回调函数运行之前运行。另一个重要的点是要知道如何正确的传递回调。这也是我常常忘记的正确语法。

无参数回调

你可以使用下面的代码来设定回调参数。

```
$.get('myhtmlpage.html', myCallBack);
```

注意 第二个参数只包含回调函数的名字（不是字符串并且不包含括号对）. Javascript 的函数可以像参数一样传递，并且在后面的代码执行。

包含参数的回调

你可能会问，"如果包含参数，你会怎么做?"。

错误的方式

错误的调用方式（不会被执行）

```
$.get('myhtmlpage.html', myCallBack(param1, param2));
```

这种方式并不会被调用，因为它调用了

myCallBack(param1, param2)

它的执行结果会被作为第二个参数传递给 $get() 函数。

更多查看原文：

http://developer.51cto.com/art/201108/281622.htm

作者：KangPeng ■

10个出色的jQuery Mobile插件

布加迪译

下面这些出色的 jQuery Mobile 插件可供你享用。它们针对移动网站进行了优化，让你可以实现诸多功能特性，比如移动滑动触摸、移动设备检测、移动浏览器检查、移动图像库、移动拖放、移动触摸滚动、移动 Ajax 调用、移动 CSS 改动及更多！尽情享用吧！

1、jQuery SwipeGallery

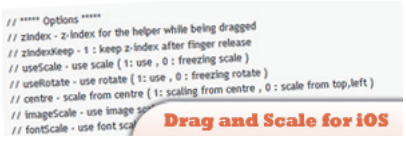


该插件针对移动网站进行了优化，支持滑动手势。它是款轻型插件：只要仅仅编写几行 HTML 代码，就可以立即创建图像库，比如排列无次序的图像列表。

相关链接：

http://plugins.jquery.com/project/swipeGallery

2、Drag and Scale for iOS



dragNscale 插件让你只要在移动平台上进行触摸和双指张合等操作，就可以让被选择的元素拖动、扩展和旋转。

相关链接：

http://plugins.jquery.com/project/dragNscale

3、Touch Scroll



这个 jQuery 插件为获得内嵌滚动内容提供了一种方法。它模拟了在 iOS 设备上滚动的行为，以便解决 “position: fixed;” 在移动版 Safari 中无法使用的问题。这个插件只适用于垂直滚动的内容。

相关链接：

http://plugins.jquery.com/project/touch-scroll

4、Keep Reading

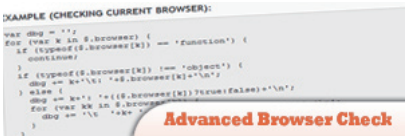


Keep Reading 插件可以用来当页面在不明显位置含有内容时，在屏幕的右下部分显示一个小小的标签。这对于移动浏览器来说非常实用，你可能希望在一个页面上有一个长长的表单，以便缩短网络行程。

相关链接：

http://plugins.jquery.com/project/keepreading

5、Advanced Browser Check



这个插件可以全面检查浏览器 (iPhone、iPad 和 iPod/ 操作系统、浏览器名称和版本 / 插件和 ActiveX 检查 /HTML5 支持 /CSS3 支持)。

相关链接：

http://plugins.jquery.com/project/advbrowsercheck

6、jQuery Canvas Loader

使用画布元素来创建典型的 ajax 装入图像

10 个出色的 jQuery Mobile 插件 II

之外的替代图像。可以用来更换现有的图像，或者你可以创建自定义图像，取代页面上的任何元素。对于不支持动画图片的浏览器来说，这个插件很有用，比如 WebKit for Android。



相关链接：

http://plugins.jquery.com/project/canvas-loader

7、jQuery Mobile Device Drag And Drop



这个项目背后的想法是，创建一个可以在普通桌面浏览器上使用的拖放实例，但又支持移动设备。这个项目旨在设计一种拖放式的界面，又支持移动设备。确切地说，大多数移动设备会“劫持”在 web 浏览器上拖动这个操作，以便提供滚动功能，这样你的拖放项在页面上保持静止。

相关链接：

http://plugins.jquery.com/project/mobiledraganddrop

8、Align Object to Bottom



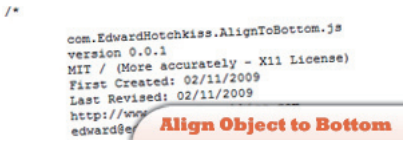
Align an Object to Bottom 使用设定的时间间隔，这样我们可以实现与 iPhone 兼容。

相关链接：

http://plugins.jquery.com/project/AlignToBottom

9、jQuery iPhone Swipe Gestures

使用面向 onTouchStart、onTouchMove、onTouch-End 和 onTouchCancel 的 Safari JavaScript 事件，支持 iPhone 和 iPod Touch 滚动手势。



相关链接：

http://plugins.jquery.com/project/swipe

10、iPhone Pinch Gestures



使用 Safari JavaScript 事件，支持 iPhone 和 iPod Touch 双指张合手势。一款新的插件（目前在开发当中）可以实现双指张合事件与任何 DOM 元素的 jQuery 绑定。

相关链接：

http://plugins.jquery.com/project/pinch

原文链接：http://mobile.51cto.com/hot-279848.htm ■

程序员小幽默

不知道身为资深程序员的你，精通下图中的哪种技术。



我们要做会写代码的项目经理

也许文章的标题起的带有讽刺的味道，其实这也是本人的一个小小的疑问。

一个项目的领导者该不该对技术有一点深度的了解或者说项目经理应该是一个不错的高级程序员。我的头跟我说项目经理不需要写代码也不需要对技术有多了解，只要对项目的进度有个整体的把控就 OK 了。这种观念一开始我不太赞同，项目经理对技术的实现没有一定的了解，在安排进度的时候是不是会草率的了事。给程序员预留的时间也不能准确的控制好，是不是会导致项目的进度控制的不太合理；

在参与开发项目的时候尤其是有一定技术含量的时候，更要项目经理对技术的实现有自己独特的见解，能帮助程序员理清头绪。但是话说回来，项目经理不可能帮每个程序员都去解决技术问题。在一些中小企业，项目经理显的很“肥胖”，这种“肥胖”完全是脱俗，对技术似乎已经到了一种炉火纯青的高度，总觉得技术无非就是增、删、改、查。这也是我的头跟我说的，程序员就是做增、删、改、查。没多少技术含量，敲来敲去都是那些东西。做为程序员的我们不太喜欢听这样的话，技术的深奥是不能用这种片面的话来概括的。

我们搞技术的，在领导看来常常有一种毛病，什么毛病呢？就是我们在解决技术问题的时候，喜欢较劲。领导会这么想也有他的道理，领导希望能把项目赶紧做完。在进度上领导永远最关心，不喜欢我们为了一个小小的技术问题，而耽误大量的时间。哪怕换一种相当麻烦的实现手段也行，保证进度第一。如果没个程序员都这样想，我担心恐怕项目的质量难以保证了。

由于项目经理的存在是随着项目的存在而存在的，我所在的开发部。项目完工后，就交给维护部门进行后期维护。经常听到维护人员的抱怨，代码写的乱、繁琐，难以维护，添加一个功能只能进行额外的编写代码，更本没办法切入代码中进行添加。

这个时候项目经理已经不复存在了，哪怕知道这个项目是某个人带的，也许这个人又带了另一个项目。这样的恶性循环，受罪的是维护人员。我们公司的业务量比较大，项目的开发周期很短，一般一个项目基本上都控制在一个月之内。为了突进度，将大量的技术不错的程序员从维护部门掉出来，进行项目的开发。在这种节骨眼上，我看到的是项目经理已经不太在乎技术实现的方法了，只在乎技术实现就行了。

基本上维护部门需要对我们开发的代码进行一个 40% 的重写，而不是重构。因为看不懂，或者说是很繁琐更本没办法维护下去。如果改出了问题，在线上报错，那么维护人员的代价就大了，轻则警告警告，重则扣钱。

所以本人一直不赞同项目经理是一个不懂技术的人，可能就是会简单的东西就带项目。很多公司都有一种现象，不关你的技术水平怎么样，只要你熟悉业务就能带项目，技术问题交给程序员，但是有没有考虑到你手下的程序员是不是大部分都是不错的。人多了不一定就能把事干好，反而让带新人的老员工分神。

一个项目经理一定要是一个不错的架构者，虽不说你精通某种技术，但最起码要求你对整个技术框架有个自己的认识。（作者：王清培）■

五分钟理解Python元类

作者 / 赖勇浩

本文源于在 PyCon UK 2008 上的一个快速演讲。让你五分钟理解 Python 元类(Metaclasses)真的,它并非巫术。

“元类的魔幻变化比 99% 的用户所担心的更多,当你搞不懂是否真的需要用它的时候,就是不需要。” —Tim Peters

元类被称为 Python 中的“深奥的巫术”。尽管你需要用到它的地方极少 (除非你基于 zope 编程),可事实上它的基础理论其实令人惊讶地易懂。

一切皆对象

- 一切皆对象
- 一切都有类型
- “class” 和 “type” 之间本质上并无不同
- 类也是对象
- 它们的类型是 type

以前,术语 type 用于内置类型,而术语 class 用于用户定义的类,但自 Pythoon 2.2 以来“class”和“type”本质上并无不同。

对于旧风格 (old-style) 类的类型是 types.ClassType。

真的,这是真的

```
Python 2.5.1 (r251:54869, Apr 18 2007, 22:08:04)
>>> class Something(object):
...     pass
...
>>> Something
<class '__main__.Something'>
 >>> type(Something)  <type 'type'>
```

从这里可以看出在交互式解释器中创建的类是一个 first class 的对象。

类的类是……

它的元类……

就像对象是类的实例一样,类是它的元类的实例。

调用元类可以创建类。

确切来说,Python 中的其它对象也是如此。

因此当你创建一个类时……

解释器会调用元类来生成它……

定义一个继承自 object 的普通类意味着调用 type 来创建它:

```
>>> help(type)
 Help on class type in module __builtin__:
class type(object)
 |  type(object) -> the object's type
 |  type(name, bases, dict) -> a new type
```

type 的第二种用法尤为重要。当 Python 解释器在执行一条类定义语句时 (如例子中最初的两行代码之后),它会用下面的参数调用 type:

- 字符串形式的类名
- 元组形式的基类序列——在我们的例子中是只有一个元素的元组 (’one-pl’)[1],如 (object,)。
- 包括由名字影射的类成员 (类属性、方法等) 的字典

简单模拟

以下代码创建了类属性的字典,然后调用 type 来创建了名为 Hello 的类。

五分钟理解 Python 元类 II

```
>>> def __init__(self):
...     self.message = 'Hello World'
...
>>> def say_hello(self):
...     print self.message
...
>>> attrs = {'__init__': __init__, 'say_hello': say_hello}
>>> bases = (object,)
>>> Hello = type('Hello', bases, attrs)
>>> Hello
<class '__main__.Hello'>
>>> h = Hello()
>>> h.say_hello()  Hello World
```

__metaclass__ 的魔法

只要在类定义中把 __metaclass__ 设置为任意有着与 type 相同参数的可调用对象，就能够提供自定义的元类。

通常使用从 type 继承的方法：

```
class PointlessMetaclass(type):
def __new__(meta, name, bases, attrs):
 # do stuff...
return type.__new__(meta, name, bases, attrs)
```

重要的是在 __new__ 方法中我们能够读取或改变传入的用以创建新类的参数。从而能够内省属性字典和改动、增加或者删除成员。

尽管当实例化一个类时这两个函数都会被调用，但覆盖 __new__ 比 __init__ 更为重要。__init__ 初始化一个实例，而 __new__ 的职责是创建它。因此如果元类用以自定义类的创建，就需要覆盖 type 的 __new__。

使用新类而非仅仅提供工厂函数的原因在于如果使用工厂函数（那样只是调用 type）的话元类不会被继承。

In Action...

WhizzBang 是一个类，但它现在已经不是 type 的实例，而是我们自定义的元类的实例了……

```
>>> class WhizzBang(object):
...     __metaclass__ = PointlessMetaclass
 ...
>>> WhizzBang
<class '__main__.WhizzBang'>
>>> type(WhizzBang)
<class '__main__.PointlessMetaClass'>
```

这有什么用？

很好的问题，元类将用在创建使用了它的新类时调用，这里是一些关于这样做的好处的观点：

◆ 装饰 (Decorate) 类的所有方法，用以日志记录或者性能剖分。

◆ 自动 Mix-in 新方法

◆ 在创建时注册类。（例如自动注册插件或从类成员创建数据库模式。）

◆ 提供接口注册功能自动发现和接口适配。

◆ 类校验：防止子类化，校验所有的方法是否都有 docstrings。

最重要之处在于元类中是在最后对 type 的调用时才真正创建类，所以可以自由地随你喜欢地改变属性字典（以及名称和元组形式的基类序列）。

一些流行的 Python ORM(Object Relational Mappers(对象关系影射)，用以和数据库协同工作）也如此使用元类。

哦，还有因为元类是继承的，所以你能够提供一个使用了你的元类的基类，而继承自它的子类就无需显式声明它了。

本文更多详细内容请查看：

http://developer.51cto.com/art/201108/281521.htm ■

HTML 5遭欧盟网络安全机构警告

欧盟计算机安全机构最近发出警告称，作为HTML5 重要组成部分的某些开发层标准正经历变更，而此类变更很可能导致大量严重安全问题的出现。

欧洲网络及信息安全局（简称 ENISA）于本周一公布了一份长达 61 页的文件，对 HTML5 以及网页编码母语的最新规范进行了详尽的分析。

HTML5 项目由万维网联盟（简称 W3C）策划并发起，该机构计划将截至本周二之前所有来自用户的建议及意见都详加考量，并酌情纳入 HTML5 的最新变更草案。最终，ENISA 在最终期限到来的前一天完成了全部意见的汇总工作。

“我认为这次的情况相当特殊，因为这可以说是大家第一次从安全的角度广泛地采纳意见，并将其引用至规格制定领域，” Giles Hogben 说道。他是 ENISA 机构安全服务部门的方案经理。

HTML5 的规范制定相当关键，因为它将成为应用程序设计人员及网页开发人员的使用指南，在未来数年中扮演举足轻重的角色。回顾过去，HTML4 使用规范自 1999 年以来就一直被频繁使用。

同理，一旦网页浏览器所使用的此类规范达不到各方面既定要求，那么每个消费者——无论是个人用户还是企业用户——都将面临巨大的安全风险。

“如今每个人都会随时随地用到浏览器，” Hogben 说道。“规范的重要性由此可见一斑。”

ENISA 对 HTML5 的总计十三项规范进行了审核，从中查出了五十一项安全问题。有些问题可以通过对固有规范进行调整加以解决，而其它一些则使得某些功能的使用存在风险，需要用户产生警觉，Hogben 说道。其中最受 ENISA 机构重视的高危功能之一名为“形式篡改”。

HTML5 规范存在于基于网页形式的“提交”按钮被放置在该网页中的任何位置上。也就是说，攻击者可以通过将其它 HTMl 注入到该页面中（例如设置另一个形式不同的‘提交’按钮），并进而导致用户信息被发送到攻击者预设的站点而非合法站点处。

这类新功能“已经使开发人员们广泛受益，” Hogben 说。“我们并不打算建议 W3C 机构将此功能取消掉，实际上只要用户在进行涉及此类应用的工作时注意到存在的潜在风险即可。”

ENISA 同样也为浏览器的运行机制征集了大量意见。举例来说，如果某位用户正在进行网上银行交易，那么当他需要在不同页面标签之间来回切换时，他至少应该使用多种不同的浏览器或者至少用到沙盒会话。

具备沙盒技术的浏览器会话能够避免其它标签——例如包含攻击页面的那类标签——利用疏漏对整个浏览器应用程序及其权限分配情况进行篡改。

ENISA 计划将全部意见和建议分类发送至 W3C 中的各对应工作团队中去，这些信息将帮助大家进一步完善将于 2012 年 1 月推出的修订版 HTML5 新规范。

译者： 核子可乐 原文：
http://developer.51cto.com/art/201108/280949.htm ■

程序员如何与非程序员一起工作

程序员在今天的科技世界中要想成功，仅有编码技能是不够的，在下面的对话中，Google 技术领导和经理 Ben Collins-Sissman(@sussman) 与 Google 工程经理 Brian Fitzpatrick(@therealfitz) 一起，讨论了与非程序员共事，维护关系的重要性。

Ben 和 Brian 说谦虚、尊重和信任是培养一个愉快的，富有成效的工作环境的重要原则，他们二人将在下周举行的 OSCON 大会上分享程序员如何很好地与其他同事一起工作。

OSCON 会议描述中提到“大规模组织操纵的艺术”，那是什么？

Brian Fitzpatrick：我认为“组织操纵”应有一个度，这意味着我对 Google 是如何运作的有较好的了解，我可以引导公司朝好的方向发展。

Ben Collins-Sussman：我怀疑 Fitz 说的是否属实，公司并不总是有一种权力结构与组织机构图匹配，它需要一些真正的调查，发现真正的权力关系所在。

Brian Fitzpatrick：据我所知，没有公司能做到权力结构和它们的组织机构图完全匹配，除非他们少于两个人。

程序员如何与非程序员同事沟通？

Ben Collins-Sussman：首先，要看他们的眼睛而不是他们的鞋子，其次，大多数人不会就纯粹的逻辑喋喋不休。

它有助于添加一些情绪的敏感性，例如，讨论时发生一些争执，大多数非工程师不会领情原始的真相，工程师喜欢纯粹的信息，往往期望十足的反馈 — 代码审查，设计方案等，当你与非技术人员沟通时，仅陈述事实并不总是有帮助，你需要以某种方式将事实包装起来，让它们更易于他人吸收。不要在你的反馈中绕圈子，确保对方不需要逐字分析你的反馈，大多数人比程序员更敏感。

Brian Fitzpatrick：尊重非程序员的能力是关键，你公司里做市场营销，销售或公关的同事，他们不知道编写代码并不意味着他们就是白痴，他们有他们的工作，你有你的工作，除了工程外，还有大量的技能需要学习，我们应该和非工程师同事相互帮助，协同工作，共同提高。

如何与其他程序员协作，有什么建议？

Brian Fitzpatrick：这是一个广泛的问题，我们的新书将会在这方面提出建议。我们主张采取三个主要的原则：谦虚、尊重和信任（简称 HRT），如果你表现得很谦虚，并尊重你的同事，并信任他们能把事情做好，你和它们建立愉快的，富有成效的工作关系的机会将更大。

当然，你的同事需要获得你的尊重和信任，但我们发现，要信任某人首先就得怀疑他，如果有人不可信任或不值得你尊重，你会很快弄明白。

Ben Collins-Sussman：这些原则不仅适用于自己，实际上可以成为一个强大团队的文化基础，我们常说软件工程是一项团队运动，而不是个人运动，围绕 HRT 培养团队文化是提高生产力，长期保持成功的关键。■

项目发生分歧时该如何处理？

更多精彩内容：

http://developer.51cto.com/art/201107/278247.htm

译者： 黄永兵

Scala登陆.NET平台



Scala 被很多人认为是下下一代的 Java，而且 Scala 被预想能在 .NET 平台上运行。今天，这个预想已经开始变成了现实。

Miguel Garcia 是瑞士洛桑理工大学的 Scala 语言开发小组成员，多年来他一直在努力将 Scala 的生产力移植到 .NET 平台，为开发者提供帮助。现在多年的工作已经达到了一个重要的里程碑。在 Miguel 的访谈中，他称已经可以在 .NET 上使用 Scala，并预计这个微软资助的重要项目将来会有更了不起的成就。

51CTO 编辑推荐：Scala 语言编程入门指南

为什么需要让 Scala 在 .NET 上运行?

Miguel: 总的来说有很多原因，这对开发者和企业两方面都非常具有吸引力。Scala 是一个精确的并且具有高度生产力的编程语言。在 .NET 上使用 Scala，开发者不仅能够更快速的开发出各种应用，而且能够在 JVM 和 .NET 这两个主流行业平台之间进行跨平台部署。

这对于开发者来说意味更多，因为只需要学习一种语言就可以在两种环境里创建应用。从企业的观点看，这也能够更好的利用珍稀资源，因为优秀的程序员可以轻松的跨越两个平台，减少了培训成本，增加了灵活性，降低了风险。.NET 平台集成了多种语言，Scala 的互操作性也很好，你可以使用现有的 .NET 库或现有的应用，不需要使用 Scala 重写。最终的效果是原先只能分别在 .NET 和 JVM 环境中使用的工具和应用都可以从一个环境移植到另一个环境，这是一个多赢的局面。从目前的情况看，Scala 和 .NET 二者结合的非常好，对 Scala 开发者来说几乎不会察觉到是在 .NET 上运行。

现在就能在 .NET 上运行 Scala 程序了吗?

Miguel: 答案很简单，是的，只是目前还有一点局限，但到今年秋天就会完全解决。

如果你想把正在 JVM 上运行的 Scala 程序移植到 .NET 上，需要做的只是使用 Scala.NET 编译器重新编译一遍，就可以在 .NET 上运行了。

现在碰到的主要局限就是 Scala 程序还不能使用 .NET 中那些用到 CLR 泛型的类库，比如 .NET 集合类。当然这只是一个小限制了，其他的功能全部可用，而且到今年秋天 .NET 泛型也能全部支持。

怎样在 .NET 上开发 Scala 程序呢?

Miguel: 你可以直接在 MS Visual Studio 里开发。到今年秋天 Visual Studio 的 Scala 插件就能支持人们期望的各种基本功能了，具体地说，IDE、代码自动完成、代码浏览、断行、表达式和异常报告等等这些。

目前的 Scala.NET 编译器也已经非常强大了，它能够对自己进行编译，大概由十万行 Scala 代码组成。现在的插件还不支持代码自动完

和代码浏览，所以我有时还是会选择使用 Eclipse 或是 Intellij 在 JVM 上编写代码，然后在 Visual Studio 中做最终的调试。

你们是怎样做到的?

Miguel: 主要需要克服的困难是 Scala 程序大量使用了 Java JDK，而 Java JDK 在 .NET 环境中是不被支持的，所以 Scala 编译器必须要针对 .NET 而重新生成代码。

Java 社区早已经开发出 IKVM 虚拟机，可以把 Java 程序转换成 MSIL（Microsoft Intermediate Language，微软中间语言），此外还有一个 .NET 类库用来提供基本的 JDK 支持。IKVM 是 Jeroen Frijters 很早以前就一直倡导的一个开源项目，也是为了遵循 Java “一次编写，到处运行” 的原则，IKVM 存在很长时间了，作为开源项目非常成功，也有可靠的商业支持。IKVM 类库是我们用来初始 Scala.NET 编译器的关键内容，而且给我们提供了将 Scala 类库搬上 .NET 的优良基础。

说来 Scala.NET 编译器的历史也很长了，但刚开始主要是集中在编译器的爱好者和黑客群体里，当时他们中有些人开发出了跨平台的编译器。在我进入的时候，跨平台编译器已经成型，但它还不能在 .NET 环境中启动自己。这意味着必须有一个能够自己编译自己的 Scala.NET 编译器，实际上这是一个先有鸡还是先有蛋的经典问题。

当时的跨平台编译器运行良好，但只能在 JDK 上运行。依靠大量的 JDK API，这和目前的 Scala 应用一样。现在很多 Scala 开发者还是习惯使用 JDK 中的功能，即使这些功能在 Scala SDK 上也已经提供了。所以说我们虽然有了跨平台编译器，但缺乏方法，能够用 .NET 类库的功能自动替换那些 JDK 的功能。

IKVM 编译器能够提取 JVM 字节码转换为 .NET 的 Assembly 类库。推进到字节码级别虽然是前进了一大步，但并没有给我们带来真正的开端，因为我们依旧没有摆脱对 JDK 的依靠。所以我们的任务就变成了创建一个前端处理器，能够修改 Scala 的源代码，将 JDK 直接替换为 IKVM 类库中的 .NET 相同功能，一种映射机制。

具体是怎样映射的呢?

Miguel: 第一步是创建源代码级所需要的初始化转换。下一阶段是扩展 Scala 编译器来应用这些转换。Scala 编译器是模块化的，支持各种插件，因此应用起来很方便，但并不是所有的编译器都能够那么灵活的做到。我们创建了一个新的 Scala 插件 JDK2IKVM 用来在批模式下完成源代码转换。得到的源代码引用 IKVM 类库，经过标准的编译器流程生成 MSIL 字节码。这就是我们现在的 Scala.NET，能够自己编译自己的 Scala 编译器。实际操作中我们会反复进行多次编译 / 转换环节来完成全部的功能转换。

在后台我们使用了重构技术，这种技术也可以用于其它的平台移植，例如将 Java.IO 自动转换为 Scala.IO。当然 Scala.NET 编译器本身也就是一个 Scala 应用，因此对于任何 JVM 上的 Scala 应用都会有效，这很好理解。

原始类型(Primitive Type)的兼容性怎样?

Miguel:.NET 的 CLR 提供的原始类型是 JVM 类型的超集，从编译器的角度看来它们转换起来很容易。字符串和数值等等都能够完成转换，从我们开始测试起就没有发现什么困难情况。IKVM 在 IKVM 社区已经使用了很多年了。■

Java 8整装待发 图谋云计算

作者 / 布加迪

鉴于Java 7 SE（标准版）现已正式发布，甲骨文和Java社区进程组织（JCP）的成员们已开始仔细考虑为这种编程语言的下一个版本Java SE 8添加什么功能特性。为这个新版本提上议程的工作是：设计面向云计算的Java。

Mark Little是红帽公司中间件事业部的高级工程主管，也是红帽针对JCP的主要联络官。他说："Java 8旨在为云计算作好准备，面向更广泛的部署领域。"他强调，为了不至于进一步推迟版本的发布，甲骨文撤掉了原计划为Java 7添加的许多高级功能特性。那些功能特性很有可能添加到Java 8中。

Little表示，结果将证明，那些功能特性中至少有两项会非常有助于让下一个版本的Java为云计算的大规模部署作好准备。一项是多租户功能，即Java虚拟机（JVM）安全地运行多个应用程序的功能。另一项是模块功能，即把Java开发工具包（JDK）重新组织成一套定义清晰但又相互关联的模块。

Little说："如果Java想在云计算环境成为主导者，那么模块功能和JVM里面真正的多租户功能对Java 8来说很重要。"

Little表示，模块功能是红帽最希望出现在Java 8中的一项特性。模块功能将减小大多数Java部署环境的规模，因为不是所有的部署环境都需要Java的全部核心库。该功能还有望帮助开发人员更容易与Java进行交互，让他们只要使用所需的部分，而不是设法应对整个代码库。

模块功能还有助于开发人员解决Little所说的"类装入器难题"（classloader hell）这个问题。

某个Java程序访问多个Java存档（JAR）即常用例程的组合时，开发人员就会遇到类装入器难题。应用程序可能会使用来自某个JAR的一个类，它实际上需要该类驻留在另一个JAR中的不同版本。或者，应用程序可能在使用由另一个程序使用的JAR；一旦那另一个程序终止，JAR就被移除，导致第一个应用程序停止运行。

Little说："为了让模块可以随意换进换出，又不破坏整个环境，就需要在JVM中同样给予支持。"

Project Jigsaw这一项计划就致力于实现这个目标。Sun公司掌控Java（Sun在2010年被甲骨文收购）时，这家公司的工程师青睐Jigsaw，而不是另一种方案：开放服务网关计划（OSGi），后者由OSGI组织监管。

Little表示，Project Jigsaw原本为Java 7而生，不过它在2010年被暂停，目的是为了在2011年之前交付Java。不过Little预测，来自Jigsaw或OSGi的工作成果都不会添加到Java 8中。他说："Java SE 8中会存在一定的模块功能。"

除了模块功能外，Java 8可能还有多租户功能，即通过一个JVM，安全地运行多个应用程序的功能。

这类功能对于Java应用于云计算环境来说必不可少；在云计算环境下，多个有关方共享同一个基础设施。

和代码浏览，所以我有时还是会选择使用 Eclipse 或是 Intellij 在 JVM 上编写代码，然后在 Visual Studio 中做最终的调试。

你们是怎样做到的?

Miguel: 主要需要克服的困难是 Scala 程序大量使用了 Java JDK，而 Java JDK 在 .NET 环境中是不被支持的，所以 Scala 编译器必须要针对 .NET 而重新生成代码。

Java 社区早已经开发出 IKVM 虚拟机，可以把 Java 程序转换成 MSIL（Microsoft Intermediate Language，微软中间语言），此外还有一个 .NET 类库用来提供基本的 JDK 支持。IKVM 是 Jeroen Frijters 很早以前就一直倡导的一个开源项目，也是为了遵循 Java “一次编写，到处运行” 的原则，IKVM 存在很长时间了，作为开源项目非常成功，也有可靠的商业支持。IKVM 类库是我们用来初始 Scala.NET 编译器的关键内容，而且给我们提供了将 Scala 类库搬上 .NET 的优良基础。

说来 Scala.NET 编译器的历史也很长了，但刚开始主要是集中在编译器的爱好者和黑客群体里，当时他们中有些人开发出了跨平台的编译器。在我进入的时候，跨平台编译器已经成型，但它还不能在 .NET 环境中启动自己。这意味着必须有一个能够自己编译自己的 Scala.NET 编译器，实际上这是一个先有鸡还是先有蛋的经典问题。

当时的跨平台编译器运行良好，但只能在 JDK 上运行。依靠大量的 JDK API，这和目前的 Scala 应用一样。现在很多 Scala 开发者还是习惯使用 JDK 中的功能，即使这些功能在 Scala SDK 上也已经提供了。所以说我们虽然有了跨平台编译器，但缺乏方法，能够用 .NET 类库的功能自动替换那些 JDK 的功能。

IKVM 编译器能够提取 JVM 字节码转换为 .NET 的 Assembly 类库。推进到字节码级别虽然是前进了一大步，但并没有给我们带来真正的开端，因为我们依旧没有摆脱对 JDK 的依靠。所以我们的任务就变成了创建一个前端处理器，能够修改 Scala 的源代码，将 JDK 直接替换为 IKVM 类库中的 .NET 相同功能，一种映射机制。

具体是怎样映射的呢?

Miguel: 第一步是创建源代码级所需要的初始化转换。下一阶段是扩展 Scala 编译器来应用这些转换。Scala 编译器是模块化的，支持各种插件，因此应用起来很方便，但并不是所有的编译器都能够那么灵活的做到。我们创建了一个新的 Scala 插件 JDK2IKVM 用来在批模式下完成源代码转换。得到的源代码引用 IKVM 类库，经过标准的编译器流程生成 MSIL 字节码。这就是我们现在的 Scala.NET，能够自己编译自己的 Scala 编译器。实际操作中我们会反复进行多次编译 / 转换环节来完成全部的功能转换。

在后台我们使用了重构技术，这种技术也可以用于其它的平台移植，例如将 Java.IO 自动转换为 Scala.IO。当然 Scala.NET 编译器本身也就是一个 Scala 应用，因此对于任何 JVM 上的 Scala 应用都会有效，这很好理解。

原始类型(Primitive Type)的兼容性怎样?

Miguel:.NET 的 CLR 提供的原始类型是 JVM 类型的超集，从编译器的角度看来它们转换起来很容易。字符串和数值等等都能够完成转换，从我们开始测试起就没有发现什么困难情况。IKVM 在 IKVM 社区已经使用了很多年了。■

编者按

程序员们总有做父亲的一天，那么究竟该怎么样在孩子们面前树立自己的权威呢？下面的小工具可能就是程序员爸爸们很好的育儿工具——Small Basic。

程序员育儿新工具Small Basic

Small Basic 1.0 基于 Basic 语言设计，当然针对小孩子们做了简化。这些未来的程序员们能在这个极简单的 IDE 开发环境里，学会最基本的 Basic 语言，还能运行调试。一切都是那么的有模有样。 51CTO 之前也曾介绍过《2011 年 6 月编程语言排行榜：为孩子创造的语言 Scratch》，看来西方人真的是编程开发从娃娃抓起。

Small Basic 下载地址：

http://down.51cto.com/data/225252

Small Basic 界面



按照惯例是 Hello World 代码登场，还是有些不一样的。

```
TextWindow.WriteLine("Hello World")
```

51CTO 资深顾问老杨，也曾是一名资深 Basic 教师。结果老杨同志在试用 Small Basic 后发来疑问，为什么写出来的代码不能编译通过。看来大家还是用的以前 Basic 的代码，在这里大家要详细阅读一个 PDF，位于 Small Basic 安装目录下。他将教会你怎么去试用 Small Basice，不过 51CTO 为了方便大家，已经将这个 PDF 放到网上，供大家下载。Small Basic 操作手册下载地址为 http://down.51cto.com/data/225254

代码编写完成后，就可以直接点击运行按钮，或者是各位程序员爸爸习惯的 F5。熟悉的 Hello World，代码编译成功。这里需要注意的是，您的机器必须安装 .NET Framework 3.5，不然编译是无法通过的。

代码写完了，Small Basic 还提供一个毕业功能。是的，你们家的小程序员该毕业了，直接把他（她）写的代码转换成标准版的 VB 代码吧。这里需要您的电脑上具备 Microsoft VB 环境，如果没有，会提示您下载免费版本。

总体来说这款 Small Basic 照顾了大部分青少年的习惯，界面设计简单，还有很详细的代码提示功能。可以说不光是孩子们玩玩的工具，普通程序员也可以用来调节身心用。但 51CTO 编辑在这里提醒各位程序员爸爸，电脑屏幕对青少年视力有大影响，还是控制他们使用电脑的时间。另外，根据 51CTO 在智联招聘获得的信息，全北京范围内 VB 程序员仅 900 个职位，还不及 PHP 和 C++ 的五分之一，进入 VB 这个行业还需谨慎。.NET 程序员真该考虑是否应该放弃 VB.NET 了。■

教你五招，提升MySQL可扩展性

在可扩展性方面，客户的要求变得越来越多，功能列表上经常会出现 20 条、50 条甚至多达 100 多条要求，但总的来说，我们可以把它们缩短为五个大类，通过五条途径来解决可扩展性问题：

1. 调整查询操作

对查询进行优化能够让你付出最少的精力就得到最多的成果。将查询功能完善的发挥出来，达到业务需求，不会被过多的流量和过重的载荷压倒。这就是为什么我们经常看见客户碰到的麻烦越来越多，随着他们网站的访问量越来越大，可扩展性的挑战也变得越来越严重，这就是问题的所在。对网站角落里那些不常用的页面做查询优化是并不必要的，那些页面并不会收到真实世界的流量。根据反映对网络应用做一定的调整是很普遍的做法，而且效果很好。

查询优化需要启用缓慢查询日志并且不断观察。使用 mk-query-digest 这个 Maatkit 套件中的强大工具来分析日志，而且要确定设置了 log_queries_not_using_indexes 标签。一旦你发现某个查询严重占用资源，那就要优化它。使用 EXPLAIN 解释机制，使用 profiler，观察索引的使用情况，创建失踪的索引，理解它是怎么进行添加和排序的。

2. 使用 Master-Master 复制

Master-Master 的 active-passive 复制模式，或者称为循环复制，不仅能带来高可用性，也能够带来高度的可扩展性。这是因为你能够即刻给你的应用分配到一块只读的从属盘。许多网络应用都按照 80/20 的规律来分割，80% 的活动用来进行读取或 SELECT，剩下的分配给 INSERT 和 UPDATE。配置你的应用或者进行重新架构，把读取需要的流量发送到从盘，这样做是可行的，这种类型的横向可扩展能力可以进一步延伸，在必要时能够附加更多块只读从盘。

3. 使用存储

这听起来是很基础的东西，也很直接，但是经常会被忽视，你至少应该确认设置了这些：

```
innodb_buffer_pool_size
key_buffer_size (MyISAM 索引缓存)
query_cache_size （使用大型 SMP 时需要小心）
thread_cache & table_cache
innodb_log_file_size & innodb_log_buffer_size
sort_buffer_size, join_buffer_size, read_buffer_size, read_rnd_buffer_size
tmp_table_size & max_heap_table_size
```

4. 磁盘读取的 RAID

你的数据库下面是什么？不知道吗，请找出来。你是在用 RAID 5 吗？

5. 调整 Key 参数

另外，有些附加参数也可以用来提高性能：

innodb_flush_log_at_trx_commit=2

它可以极大的提升 insert 和 update 的速度，只是在清除 innodb 日志缓冲区时有点偷懒。你可以对它多做些研究。更多内容请看原文：

http://database.51cto.com/art/201108/280623.htm ■

编者按

Redis 在很多方面与其他数据库解决方案不同：它使用内存提供主存储支持，而仅使用硬盘做持久性的存储；它的数据模型非常独特，用的是单线程。另一个大区别在于，你可以在开发环境中使用 Redis 的功能，但却不需要转到 Redis。

几点建议，让Redis发挥更大作用

Redis 在很多方面与其他数据库解决方案不同：它使用内存提供主存储支持，而仅使用硬盘做持久性的存储；它的数据模型非常独特，用的是单线程。另一个大区别在于，你可以在开发环境中使用 Redis 的功能，但却不需要转到 Redis。

转向 Redis 当然也是可取的，许多开发者从一开始就把 Redis 作为首选数据库；但设想如果你的开发环境已经搭建好，应用已经在上面运行了，那么更换数据库框架显然不那么容易。另外在一些需要大容量数据集的应用，Redis 也并不适合，因为它的数据集不会超过系统可用的内存。所以如果你有大数据应用，而且主要是读取访问模式，那么 Redis 并不是正确的选择。

然而我喜欢 Redis 的一点就是你可以把它融入到你的系统中来，这就能够解决很多问题，比如那些你现有的数据库处理起来感到缓慢的任务。这些你就可以通过 Redis 来进行优化，或者为应用创建些新的功能。在本文中，我就想探讨一些怎样将 Redis 加入到现有的环境中，并利用它的原语命令等功能来解决传统环境中碰到的一些常见问题。在这些例子中，Redis 都不是作为首选数据库。

显示最新的项目列表

下面这个语句常用来显示最新项目，随着数据多了，查询毫无疑问会越来越慢。

```
SELECT * FROM foo WHERE ... ORDER BY time DESC LIMIT 10
```

在 Web 应用中，“列出最新的回复”之类的查询非常普遍，这通常会带来可扩展性问题。这令人沮丧，因为项目本来就是按这个顺序被创建的，但要输出这个顺序却不得不进行排序操作。

类似的问题就可以用 Redis 来解决。比如说，我们的一个 Web 应用想要列出用户贴出的最新 20 条评论。在最新的评论边上我们有一个“显示全部”的链接，点击后就可以获得更多的评论。

我们假设数据库中的每条评论都有一个唯一的递增的 ID 字段。

我们可以使用分页来制作主页和评论页，使用 Redis 的模板：

- 每次新评论发表时，我们会将它的 ID 添加到一个 Redis 列表：

```
LPUSH latest.comments <ID>
```

- 我们将列表裁剪为指定长度，因此 Redis 只需要保存最新的 5000 条评论：

```
LTRIM latest.comments 0 5000
```

- 每次我们需要获取最新评论的项目范围时，我们调用一个函数来完成（使用伪代码）：

```
FUNCTION get_latest_comments(start,num_items):
```

```
                id_list = redis.lrange(&quot;latest.
comments&quot;,start,start+num_items-1)
            IF id_list.length &lt; num_items
                id_list = SQL_DB(&quot;SELECT ...
ORDER BY time LIMIT ...&quot;)
            END
            RETURN id_list
        END
```

这里我们做的很简单。在 Redis 中我们的最新 ID 使用了常驻缓存，这是一直更新的。但是我们做了限制不能超过 5000 个 ID，因此我们的获取 ID 函数会一直询问 Redis。只有在 start/count 参数超出了这个范围的时候，才需要去访问数据库。

我们的系统不会像传统方式那样“刷新”缓存，Redis 实例中的信息永远是一致的。SQL 数据库（或是硬盘上其他类型数据库）只是在用户需要获取“很远”的数据时才会被触发，而主页或第一个评论页是不会麻烦到硬盘上的数据库了。

删除与过滤

我们可以使用 LREM 来删除评论。如果删除操作非常少，另一个选择是直接跳过评论条目的入口，报告说该评论已经不存在。

有些时候你想要给不同的列表附加上不同的过滤器。如果过滤器的数量受到限制，你可以简单的为每个不同的过滤器使用不同的 Redis 列表。毕竟每个列表只有 5000 条项目，但 Redis 却能够使用非常少的内存来处理几百万条项目。

排行榜相关

另一个很普遍的需求是各种数据库的数据并非存储在内存中，因此在按得分排序以及实时更新这些几乎每秒钟都需要更新的功能上数据库的性能不够理想。

典型的比如那些在线游戏的排行榜，比如一个 Facebook 的游戏，根据得分你通常想要：

- 列出前 100 名高分选手
- 列出某用户当前的全球排名

这些操作对于 Redis 来说小菜一碟，即使你有几百万个用户，每分钟都会有几百万个新的得分。

模式是这样的，每次获得新得分时，我们用这样的代码：

```
ZADD leaderboard <score> <username>
```

你可能用 userID 来取代 username，这取决于你是怎么设计的。

得到前 100 名高分用户很简单：ZREVRANGE leaderboard 0 99。

用户的全球排名也相似，只需要：ZRANK leaderboard <username>。

按照用户投票和时间排序

排行榜的一种常见变体模式就像 Reddit 或 Hacker News 用的那样，新闻按照类似下面的公式根据得分来排序：

```
score = points / time^alpha
```

因此用户的投票会相应的把新闻挖出来，但时间会按照一定的指数将新闻埋下去。下面是我们的模式，当然算法由你决定。

现在你就可以用上 Redis，让用户感到更轻松，让你的系统变得不再复杂，让你的网站反应更快。你不需要改变现有的数据库结构，使用 Redis 给你的框架带来新的东西，来完成那些从前认为不可能做到 / 很难做到的，或是成本太高的任务。

未完，详情请看：

http://database.51cto.com/art/201107/276333.htm ■

YiBo微博:程序员创业之梦



YiBo 微博是一个纯手机版的应用,纯粹到什么地步?纯粹到它的官方网站简单到无以附加的地步。下文就采访了 YiBo 微博的创始人,谈程序员的创业之梦。

YiBo 微博是一个纯手机版的应用,纯粹到什么地步?纯粹到它的官方网站简单到无以附加的地步:访问 www.dev123.net 这个 YiBo 微博的官方网站,你会发现没有任何页面设计,十几行文字介绍,一个下载链接,再无其他。然而就是这个在电脑浏览器上简陋的无以附加的 YiBo 微博,却成为了在 Android 手机系统上最炙手可热的多帐号微博工具。

一,汤奕敏和马庆升

汤奕敏和马庆升是大学同学,今年刚刚 25 岁,走出校园也刚 3 个年头。在这个时代,有自信的开发者都有着强烈的创业愿望,汤奕敏和马庆升也不例外。不过刚刚走出校园的他们还没有一个清晰的发展方向,两人决定先工作,等待机会。

汤奕敏 2008 年毕业后进入北京金山软件从事游戏的运营开发,马庆升则进入了厦门新科技公司,参与移动电信项目的开发工作。两个人没有想到的是,机会很快的到来了,从 2009 年开始,在 iPhone 的影响和 Android 系统手机的迅速发展下,中国智能手机出现了爆发式的增长,伴随着智能手机终端的普及,智能手机用户需要大量的手机专用软件,一个个基于苹果 IOS 系统和 Google 的 Android 系统的应用软件开发团队、公司开始成为了 2009 年以来移动互联网领域上热门的获得投资的方向。2009 年下半年,新浪微博上线以来微博用户增长迅速,汤奕敏和马庆升,一个在北京,一个在厦门,他们不约而同的发现了这个容易入手、拥有大量用户基础且目标明确的创业产品方向:手机上的第三方微博客户端。

在 2010 年,国外的微博 twitter 已经开始大红大紫,基于 Twitter 应用的工具软件非常多,而当时针对国内四大微博平台(新浪、腾讯、网易、搜狐)的工具软件还比较少,市场上体验比较好的工具就更少了。这让汤奕敏和马庆升觉得,他们的机会来了:产品上有国外的好经验可以借鉴,市场上国内还相对空白。

两个年轻人没有多犹豫:“走,干吧!”

二,“我们需要鼓励!”

汤奕敏和马庆升都是 Google 产品的爱好者,作为开发者他们对 Google 产品有天生的亲近感。两个人对 Android 平台都很熟悉,当汤奕敏辞掉金山软件的工作,南下厦门和马庆升开始他们自己的创业之路的时候,他们把全部精力和时间都投入到他们的新产品上。很快,YiBo 上线了。YiBo 微博是 Android 平台上的多微博客户端,同时支持新浪、腾讯、搜狐、网易;对方许多用户厌倦官方客户端相对传统和呆板的用户来说,YiBo 是一个新的选择,易用的界面,良好的用户体验。

YiBo 微博：程序员的创业之梦 II

同时可多微博阅读和同步，对于多微博使用者来说，YiBo 更是一款不可多得的必备软件。

汤奕敏和马庆升把 YiBo 提交到 Android 电子市场中，供 Android 手机用户的下载安装。两个程序员，没有花一分钱推广费用，没有做任何广告，他们看到他们的产品开始有用户下载使用，开始有用户给他们的产品写下评论……很快，他们的 YiBo 已经有超过 10 万人下载使用，对于两个刚刚开始独立创业的程序员来说，这是一个很了不起的成就。汤奕敏说，他至今还记得他在 Android 电子市场上看到第一个好评时的情景，当时一个用户给他们的 YiBo 打了五颗星的评价（最高级别），并写留言道："这是我用过的最好的微博客户端。"汤奕敏和马庆升高兴的像刚刚得到圣诞礼物的孩子一样跳起来。对于这样两个纯粹的程序员来说，没有公关，没有水军，一切都是那么真切的用户回馈，这让他们兴奋不已，"一句用户的好评甚至比 10000 块钱的收入更能激励我们做好这个产品。"汤奕敏说道，"经常有用户回复我们说感谢我们做出 YiBo 这个微博工具，其实我们应该感谢他们，没有他们的鼓励我们可能做不了这么好。对于创业者，我们需要用户鼓励！"

汤奕敏说："我们觉得，既然我们需要支持和鼓励，其实用户也一样需要肯定和赞许。我们努力回答每一个用户给我们提出的建议，意见甚至批评，我们感谢他们每一次对我们的鼓励。我们要让用户感觉到他们不是在和冷冰冰的'官方帐号'在打交道，而是在和一个真正的人在交流。"话是这么说的，他们也是这么做的。汤奕敏和马庆升每天工作 16 个小时以上，除了写代码，改进产品，就把所有业余时间放在微博平台上和他们的用户交流。

三，YiBo 的困难和未来

2011 年 5 月 10 日下午，YiBo 的用户像往常一样在手机上打开 YiBo，突然发现他们不能使用 YiBo 来管理和使用新浪微博了。一时间无数用户的求救信息发了过来。新浪微博是中国最大的微博平台，用户数也是最多的，如果 YiBo 无法打通新浪微博，那么 YiBo 还有多少用户能留下来这个是很难想象的。汤奕敏和马庆升连晚饭都顾不上吃，全力去解决这个问题。"困难总是很多的，这个我们心里有准备。但是如果是技术问题，我们总能想办法解决，然而这次事故却和我们 YiBo 的技术无关，是因为我们 YiBo 可以同时同步把用户想说的话一键同步发送到新浪、腾讯、网易和搜狐四个微博平台上，新浪微博就突然给我们 YiBo 介入他们的系统的接口降权了。这导致我们的接口调用次数远远不足，于是我们的用户就无法使用 YiBo 来登录、使用新浪微博了。"后来努力下新浪微博终于恢复了 YiBo 的权限。

YiBo 团队到现在只有 4 个人，两个核心程序员，一个产品经理，一个负责推广。比较起团队，汤奕敏和马庆升更缺乏资金，YiBo 从一开始的资金就来源于两个年轻的创始人自己的投入。缺乏 VC 的投入让 YiBo 的成长也显得不够快，半年时间，20 万用户。"如果能够有更多的支持，以目前 YiBo 的产品体验和用户口碑，我么一定可以迅速突破每天 100 万登录用户。"汤奕敏对自己的产品显得非常有信心。

原文链接：
http://developer.51cto.com/art/201107/278333.htm ■

【专家专栏】李洋谈Android安全框架

移动开发正在成为软件领域的发展趋势，51CTO 非常荣幸邀请到安全专家李洋为我们开设的专家专栏，李洋老师这个专栏里将向我们介绍“Android 安全框架”等内容，希望读者喜欢。



本文李洋老师将向我们介绍“Android 安全框架概览”：Android 安全结构的中心思想为“应用程序在默认的情况下不可以执行任何对其他应用程序，系统或者用户带来负面影响的操作。”开发者只有了解到 Android 的安全架构才能设计出在使用过程中更加流畅的用户体验。

1. Android 具有哪些权限

根据用户的使用过程体验，可以将 Android 涉及的权限大致分为如下三类：

(1)Android 手机所有者权限：自用户购买 Android 手机（如 Samsung GT-i9000）后，用户不需要输入任何密码，就具有安装一般应用软件、使用应用程序等的权限；

(2)Android root 权限：该权限为 Android 系统的最高权限，可以对所有系统中文件、数据进行任意操作。出厂时默认没有该权限，需要使用 z4Root 等软件进行获取，然而，并不鼓励进行此操作，因为可能由此使用户失去手机原厂保修的权益。同样，如果将 Android 手机进行 root 权限提升，则此后用户不需要输入任何密码，都将能以 Android root 权限来使用手机。

(3)Android 应用程序权限：Android 提供了丰富的 SDK(Software development kit)，开发人员可以根据其开发 Android 中的应用程序。而应用程序对 Android 系统资源的访问需要有相应的访问权限，这个权限就称为 Android 应用程序权限，它在应用程序设计时设定，在 Android 系统中初次安装时即生效。

值得注意的是：如果应用程序设计的权限大于 Android 手机所有者权限，则该应用程序无法运行。如：没有获取 Android root 权限的手机无法运行 Root Explorer，因为运行该应用程序需要 Android root 权限。

2. Android 组件模型（Component Model）

Android 系统中包括 4 种组件

(1)Activity：Activity 就是一个界面，这个界面里面可以放置各种控件。比如：Task Manager 的界面、Root Explorer 的界面等；

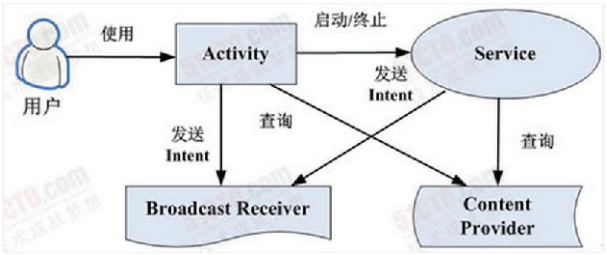
(2)Service：服务是运行在后台的功能模块。如文件下载、音乐播放程序等；

(3)Content Provider：它是 Android 平台应用程序间数据共享的一种标准接口，它以类似于 URI(Universal Resources Identification) 的方式来表示数据，如：content://contacts/people/1101;

(4)Broadcast Receiver：与此组件相关的概念是 Intent，这是一个对动作和行为的抽象描述，负责组件和程序之间的消息传递。Broadcast Receiver 组件则提供了一种把 Intent 作为一个

广播出去，由所有对其感兴趣的程序对其作出反应的机制。举个简单的例子，：为了实现一个系统启动后播放音乐的功能，则可以定义 Intent 为 android.intent.action.BOOT_COMPLETED，由 Broadcast Receiver 组件将其进行广播，而系统中 Media Player 接收到该信息后则进行播放。

如上所述，4 个组件之间的关系如下图：



4 个组件之间的关系图

3. Android 安全访问设置

每个 Android 的 apk(Android Package) 包里面都包含有一个 AndroidMainifest.xml 文件，该文件除了罗列应用程序运行时库、运行依赖关系等之外，还会详细地罗列出该应用程序所需的系统访问。该文件的基本格式如下：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.
android.com/apk/res/android"
package="cn.com.fetion.android"
android:versionCode="1"
android:versionName="1.0.0">
<application android:icon="@drawable/icon"
android:label="@string/app_name">
<activity android:name=".welcomActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.
action.MAIN" />
<category android:name="android.intent.
category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-permission android:name="android.
permission.SEND_SMS"></uses-permission>
</manifest>
```

如上述文件描述中斜体部分，该文件的作用是说明该软件具备发送短信的功能。Android 定义了 106 种 permission，可供开发人员使用。

【作者简介】

李洋博士毕业于中科院计算所。10 多年来一直从事计算机网络信息安全研发工作，曾主持和参与多项国家重点项目以及信息安全系统和企业信息安全系统的研发工作。具有 Linux 系统应用、管理、安全及内核的研发经验，擅长网络安全技术、协议分析、Linux 系统安全技术、Linux 系统及网络管理、Linux 内核开发等。

http://mobile.51cto.com/hot-280012.htm ■

《Linux 运维趋势》是由 51CTO 系统频道策划、针对 Linux/Unix 系统运维人员的一份电子杂志，内容从基础的技巧心得、实际操作案例到中、高端的运维技术趋势与理念等均有覆盖。

地址 http://os.51cto.com/art/201108/283160.htm





数据库云：轮回就是 File 到 RDBMS 到 NoSQL 的过程。自数据库产生以来，主要经历了四个阶段：人工管理阶段；文件系统阶段；数据库系统阶段；高级数据库阶段。除去人工管理阶段，轮回说，是否说明数据库再次回到文件管理阶段？那么这是进化吗？怎么看都像是在退化，倒退！想查看更多内容，请访问数据库的“轮回”专题，链接：http://database.51cto.com/database/lunhuishangsheng/

对 SQL 说不！ NoSQL 的数据库技术革命

http://database.51cto.com/art/200907/134015.htm

就像当年波士顿的爱国者为反抗英国重税一样，NoSQL 的支持者们从各地涌来，分享他们如何推翻缓慢而昂贵的关系数据库的暴政，开始对 SQL 说不！

数据库发展阶段之文件系统阶段

http://database.51cto.com/art/201107/278636.htm

1956 年 IBM 生产出第一个磁盘驱动器—— the Model 305 RAMAC。此驱动器有 50 个盘片，每个盘片直径是 2 英尺，可以储存 5MB 的数据。

数据库发展阶段之数据库系统阶段

http://database.51cto.com/art/201107/278638.htm

60 年代后，随着计算机在数据管理领域的普遍应用，人们对数据管理技术提出了更高的要求：希望面向企业或部门，以数据为中心组织数据，减少数据的冗余，提供更高的数据共享能力。